

**Determinar los requisitos funcionales y no funcionales del software de acuerdo con los requerimientos del cliente.**

**Aprendiz:**

Manuel Prudencio Pertuz Pérez

Servicio Nacional de Enseñanza – SENA

Centro Minero Regional Boyacá.

2977343 – Análisis Y Desarrollo de Software.

**Instructor:**

Andrés Mauricio Blandón Sánchez

Noviembre - 2024.

## **Tabla de Contenido**

<b>Documento de Especificación de Requisitos de Software</b>	<b>3</b>
<b>1. Introducción</b>	<b>3</b>
1.1 Propósito	3
1.2 Alcance	4
<b>2. Requisitos funcionales</b>	<b>4</b>
2.1 Módulo administrativo	4
2.2 Módulo web	5
<b>3. Requisitos no funcionales</b>	<b>5</b>
<b>4. Técnicas utilizadas para el análisis de requisitos</b>	<b>6</b>
<b>5. Coherencia entre requisitos y software a construir</b>	<b>6</b>
<b>6. Conclusión</b>	<b>7</b>

# Documento de Especificación de Requisitos de Software

**Nombre del proyecto:**

Software de Gestión y Página Web para la Academia de Música

**Versión:**

1.0

**Fecha:**

Noviembre 2024

**Autor:**

Manuel Pertuz Pérez

## 1. Introducción

### 1.1 Propósito

Este documento define los requisitos funcionales y no funcionales para el desarrollo de un software que permita gestionar las operaciones de una academia de música. Incluye funcionalidades como registro de estudiantes, consulta de horarios, gestión de pagos y generación de reportes, así como el diseño de una página web para mejorar la visibilidad en línea.

## 1.2 Alcance

El sistema ofrecerá:

- Una plataforma administrativa para gestionar estudiantes, profesores, clases y pagos.
- Una página web informativa con un módulo para que los usuarios puedan consultar horarios y registrar solicitudes.

Este software busca optimizar la operación interna de la academia y mejorar la experiencia de estudiantes, padres y profesores.

## 2. Requisitos funcionales

Los **requisitos funcionales** describen las capacidades específicas que el sistema debe ofrecer.

### 2.1 Módulo administrativo

#### 1. Registro de estudiantes

- El sistema permitirá al administrador registrar estudiantes con los siguientes datos:
  - Nombre completo.
  - Edad.
  - Nivel musical.
  - Curso asignado.

#### 2. Gestión de horarios y clases

- El sistema permitirá asignar horarios a los profesores y estudiantes.
- Se notificará automáticamente a los usuarios sobre cambios en horarios.

#### 3. Gestión de pagos

- Registro de pagos realizados por los estudiantes.
- Generación de recibos en formato digital.

#### **4. Generación de reportes**

- Crear reportes de asistencia, pagos y progreso académico de los estudiantes.

### **2.2 Módulo web**

#### **1. Consulta de disponibilidad de profesores y horarios**

- Los usuarios podrán buscar horarios disponibles para clases específicas.

#### **2. Registro de solicitudes**

- Permitir que estudiantes o padres soliciten información sobre cursos disponibles.

#### **3. Visualización de información general**

- Mostrar información sobre cursos, profesores y precios.

### **3. Requisitos no funcionales**

Los **requisitos no funcionales** establecen las características de calidad y restricciones del sistema.

#### **1. Rendimiento**

- El sistema debe responder en menos de 2 segundos para consultas básicas, como buscar horarios.

#### **2. Escalabilidad**

- La aplicación debe soportar hasta 500 usuarios simultáneos.

#### **3. Seguridad**

- Los datos de los estudiantes y profesores deben estar protegidos con cifrado.
- Acceso restringido por roles (administradores, estudiantes, profesores).

#### **4. Usabilidad**

- La interfaz debe ser intuitiva, con botones y menús claros.
- Proporcionar tutoriales básicos para el administrador.

## 5. **Compatibilidad**

- Compatible con navegadores modernos (Chrome, Firefox, Edge).
- Soporte para dispositivos móviles y tablets.

## 6. **Disponibilidad**

- El sistema debe estar disponible el 99.9% del tiempo.

## 7. **Mantenimiento**

- El código debe seguir estándares para facilitar futuras actualizaciones.

## 4. **Técnicas utilizadas para el análisis de requisitos**

### 1. **Análisis preliminar de stakeholders:**

Se analizaron los roles involucrados (administradores, estudiantes, padres y profesores) para entender sus necesidades.

### 2. **Priorización MoSCoW:**

Los requisitos fueron categorizados como:

- **M (Must have):** Requisitos esenciales para el funcionamiento.
- **S (Should have):** Requisitos importantes pero no críticos.
- **C (Could have):** Requisitos deseables.
- **W (Won't have for now):** Requisitos fuera del alcance inicial.

### 3. **Técnica de puntos de historia:**

Se estimó la complejidad de cada funcionalidad para su planificación.

### 4. **Estándar IEEE 830:**

Este documento sigue el formato recomendado para especificación de requisitos.

## 5. **Coherencia entre requisitos y software a construir**

Los requisitos funcionales y no funcionales están alineados con el propósito del software, que busca:

- Automatizar procesos administrativos como registro de estudiantes y gestión de pagos.
- Mejorar la experiencia del usuario mediante un portal web informativo.

## **6. Conclusión**

Este documento provee un marco inicial para la construcción del sistema, priorizando funcionalidades críticas para el éxito de la academia. Una vez realizada la fase de campo, se validarán y ajustarán estos requisitos para garantizar que el software cumpla con las expectativas reales de los usuarios.